University of Minnesota
Department of Computer Science
CSci 5103 - Spring 2018 (Instructor: Tripathi)
Midterm Exam 2 — Date: April 11, 2018
CLOSED BOOK/NOTES
(Time: 75 minutes) Total Points – 100

STUDENT NAME:
STUDENT ID:

Problem 1	Problem 2	Problem 3	Problem 4	Problem 5	Problem 6	Total Score
15	15	15	10	25	20	100

Problem 1: (15 points) For a system, the resource allocation state is given below:

Available = (2 4 1 1)

Process	Allocation	Remaining Need
P0	1 2 0 0	2 2 5 1
P1	0 0 1 1	4 5 2 2
P2	1 0 0 2	1 2 1 1
P3	1 0 2 0	2 2 0 1
P4	2 2 1 3	2 2 1 3

(a) (10 points) Is this system in a safe state? Justify your answer. If the system state is safe, give a completion sequence for the processes.

(b) (5 points) If a request from process P0 arrives for (0 2 1 0), can the request be immediately granted? Justify your answer.

continue answer for Problem 1...

Problem 2: (15 points) It has been observed that the number of instructions executed between page faults is directly proportional to the number of page frames allocated to the program. If the allocated memory is doubled, the mean interval between page faults is also doubled. Suppose that a normal instruction takes 1 microsecond, but if a page fault occurs, it takes 2001 microseconds (i.e. 2 milliseconds for page fault handling). If a program takes 60 seconds to run, during which it gets 15,000 page faults, how long would it take to run if twice as much memory were allocated to it?

Problem 3: (15 points) A computer has four page frames. The time of loading, time of last access, and the R and M bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last ref.	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

1. Which page will NRU replace?
2. Which page will FIFO replace?
3. Which page will LRU replace?
4. Which page will second chance replace?

Problem 4: (10 points) Suppose that the WSClock page replacement algorithm uses a $\tau = 2$ ticks and the system state is given below. You may assume any position for the clock-hand. The three flag bits V, R, and M stand for Valid, Referenced, and Modified, respectively.

Page	Time stamp	V	R	M
0	6	1	0	1
1	9	1	1	0
2	9	1	1	1
3	7	1	0	0
4	4	0	0	0

1. If a clock interrupt occurs at tick 10, show the contents of the new table entries. Explain (You can omit entries that are unchanged.)
2. Suppose that instead of a clock interrupt, a page fault occurs at tick 10 due to a read request to page 4. Show the contents of the new table entries. Explain. (You can omit entries that are unchanged.)

Problem 5: (25 points) (5 points for each part)

Part A: (5 points) Suppose that a computer machine architecture supports page-tables with protection-bits for virtual memory management, but it does not provide the *reference bit R* and *dirty bit M* at the hardware level. Briefly outline how one can simulate the functions of these bits using software methods.

Part B (5 points) Page locking prevents a memory page from being replaced at a particular point in time. Give at least two examples where this mechanism is needed by the kernel.

Part C (5 points) Some operating systems maintain a pool of free frames and when a page-fault occurs, the kernel *borrows* one frame from the pool and initiates the disk operation to read the demanded page in that frame. When a victim page for replacement is found, its frame is added to the free pool. Identify two advantages of this approach.

Part D (5 points) For supporting memory virtualization, using a simple example illustrate the distinction between **guest-induced page-faults** and **hypervisor-induced page-faults**.

Part E (5 points) In memory virtualization, what is a “shadow page-table”? How does the Virtual Machine Monitor ensure the consistency of shadow page tables when the guest OS operating system modifies the page-table of a process following its own page replacement/allocation policies.

Problem 6: (20 points)
The virtual memory space used by a program has 5K words. This program is executed on two different machines, one with 1K word size pages and the other with 512 words size pages. Thus in the first system the program's address space has five pages and in the second system it has 10 pages. Each logical page in the first system splits into two logical pages in the second system. For example page number 1 in the first system splits into two pages numbered $1'$ and 1" in the second system, page 2 splits into $2'$ and 2", and similarly for the other pages.

In both systems, the program is given 4K words of physical memory. (There are 4 page-frames in the first system, and 8 page-frames in the second system.) The memory reference strings generated by the program in the two systems have the following form (these two are equivalent in terms of accessing locations in the virtual address space):

System I: $(1\ 2\ 3\ 4\ 5\ 5\ 4\ 3\ 2\ 1)^N$ where N is a large integer
System II:(1' 2' 3' 4' 5' 5" 4" 3" 2" 1")^N

(a) (16 points) Assuming LRU replacement policy, for some given value of N, give the number of page-faults that will occur in each of these two systems.

(b) (4 points) Assume that handling a page-fault takes the same amount of time on both these systems. How do the execution times for this program on these two systems compare? Which system will finish first? Justify your answer.

continue answer for Problem 6...